Phase 2: Exporting Data

First, let's discuss exporting data. We previously talked about the Serial Monitor, where all data is real-time, but it cannot be saved. For real-time data, the purpose of using the data is for subsequent planning, analysis, and application. Therefore, saving data is also a crucial step. Here, we will store the data in Excel CSV format.

Required Python Libraries:

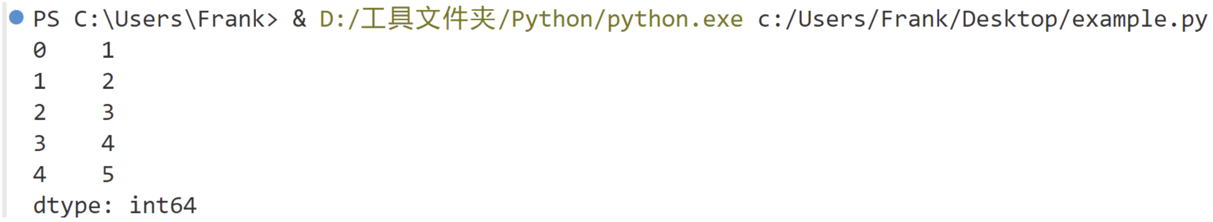
PySerial: PySerial is a Python library used for communication through serial ports (like COM ports). It allows data exchange with hardware devices connected to serial ports, such as Arduino and sensors. PySerial is essential for real-time communication, enabling Arduino to send all data to Python in real-time. Arduino handles basic data processing, while Python can perform more advanced tasks. To install PySerial, use the following command: pip install pyserial.

Pandas: Pandas is a powerful data processing and analysis library. It provides convenient operations for various data structures, with two core data structures: ‘Series’ and ‘DataFrame’.

1. Series: A one-dimensional array similar to a Python list.

```
import pandas as pd
data = [1, 2, 3, 4, 5]
series = pd.Series(data)
print(series)
```

output

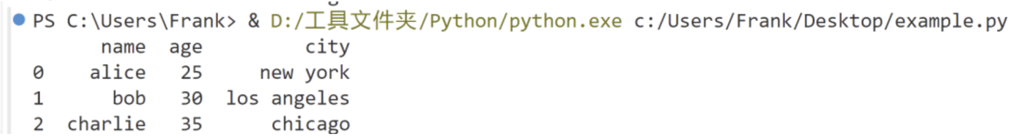


```
PS C:\Users\Frank> & D:/工具文件夹/Python/python.exe c:/Users/Frank/Desktop/example.py
0    1
1    2
2    3
3    4
4    5
dtype: int64
```

2. DataFrame: A two-dimensional table data structure, similar to Excel. Run these lines of code, which are very similar to the format in Excel, and can operate on the data. One is that the filter age must be greater than 30, and the other is that the sort.

```
import pandas as pd
data = {
    'name':['alice','bob','charlie'],
    'age':[25,30,35],
    'city':['new york','los angeles', 'chicago']
}
df = pd.DataFrame(data)
print(df)
```

output



```
PS C:\Users\Frank> & D:/工具文件夹/Python/python.exe c:/Users/Frank/Desktop/example.py
      name  age         city
0    alice   25     new york
1      bob   30  los angeles
2  charlie   35      chicago
```

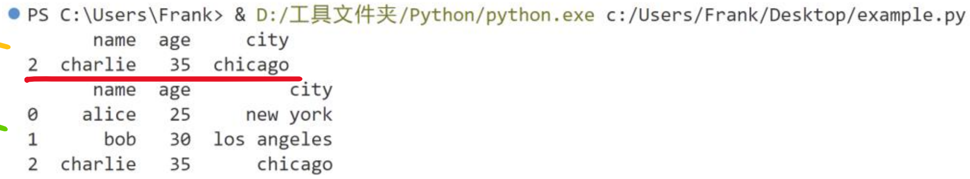
Filtering and sorting data:

```
import pandas as pd
data = {
    'name':['alice','bob','charlie'],
    'age':[25,30,35],
    'city':['new york','los angeles', 'chicago']
}
df = pd.DataFrame(data)
#print(df)

df_filtered = df[df['age']>30] # filter age must be greater than 30
print(df_filtered)

df_sorted = df.sort_values(by='age') # the sort
print(df_sorted)
```

output



```
PS C:\Users\Frank> & D:/工具文件夹/Python/python.exe c:/Users/Frank/Desktop/example.py
      name  age     city
2  charlie   35  chicago
      name  age         city
0    alice   25     new york
1      bob   30  los angeles
2  charlie   35      chicago
```

The key difference between ‘Series’ and ‘DataFrame’ is that ‘Series’ is one-dimensional while ‘DataFrame’ is two-dimensional. Pandas is crucial for data processing and is one of Python's core libraries.

Openpyxl: Openpyxl is a library used for reading and writing Excel files (.xlsx format). It provides a range of functions to manipulate Excel spreadsheets. To install Openpyxl, use the following command: pip install openpyxl.

```
import serial  # For serial communication
import pandas as pd  # For data processing
import time  # For timestamps and loop delay

# Open serial communication on COM8 with a baud rate of 9600
ser = serial.Serial('COM8', 9600)

# Initialize a dictionary to store timestamps, humidity, and temperature data
data = {'Timestamp': [], 'Humidity': [], 'Temperature': []}

try:
    while True:
        line = ser.readline().decode('utf-8').strip()  # Read serial data
        # ser.readline() reads a line of data, decode('utf-8') decodes it to a
UTF-8 string, strip() removes whitespace

        if "Humidity" in line and "Temperature" in line:  # Check if line
contains "Humidity" and "Temperature"
            parts = line.split('\t')
            humidity = parts[0].split(': ')[1].split(' ')[0]  # Extract humidity
value
            temperature = parts[1].split(': ')[1].split(' ')[0]  # Extract
temperature value

            # Add timestamp, humidity, and temperature data to dictionary
            data['Timestamp'].append(time.strftime("%Y-%m-%d %H:%M:%S"))
            data['Humidity'].append(humidity)
            data['Temperature'].append(temperature)

            print(f'Time: {data["Timestamp"][-1]}, Humidity: {humidity},
Temperature: {temperature}')

            # Save data to Excel file every 10 entries
            if len(data['Timestamp']) % 10 == 0:
                df = pd.DataFrame(data)  # Convert dictionary to DataFrame
                df.to_excel('sensor_data.xlsx', index=False)  # Save DataFrame to
Excel file without index
                print('Data saved to Excel')

        time.sleep(2)  # Pause for 2 seconds to control data reading frequency

# Save data to Excel file and close serial port on program exit
except KeyboardInterrupt:
    df = pd.DataFrame(data)
    df.to_excel('sensor_data.xlsx', index=False)
    ser.close()
    print('Data saved to Excel and serial port closed')
```

This code reads data from the Arduino via the serial port, processes it using Pandas, and saves it to an Excel file using Openpyxl. It demonstrates real-time data acquisition, processing, and storage.